ARTIFICIAL INTELLIGENCE

(CSC 462)

LAB ASSIGNMENT # 1

NAME: MUAAZ BIN MUKHTAR

REG NO: FA21-BSE-045

CLASS & SECTION: BSSE-5A

SUBMITTED TO: SIR WAQAS ALI

DATE SUBMITTED: 23-12-2023

Department of Computer Science

QUESTION 1

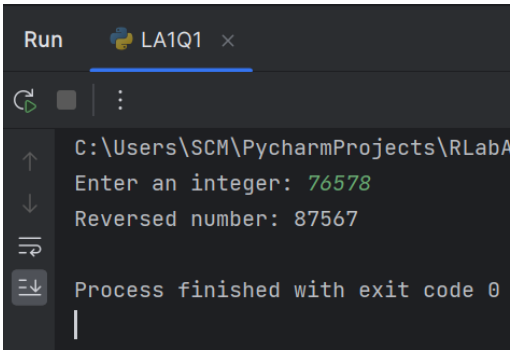
Write a program that prompts the user to input an integer and then outputs the number with the digits reversed.
For example, if the input is 12345, the output should be 54321.

Answer:

Code:

```
user_input = input("Enter an integer: ")

if user_input.isdigit():
    reversed_number = int(str(user_input)[::-1])

    print("Reversed number:", reversed_number)
else:
    print("Please enter a valid integer.")
```

Output:



Question No. 2:

Write a program that reads a set of integers, and then prints the sum of the even and odd integers.

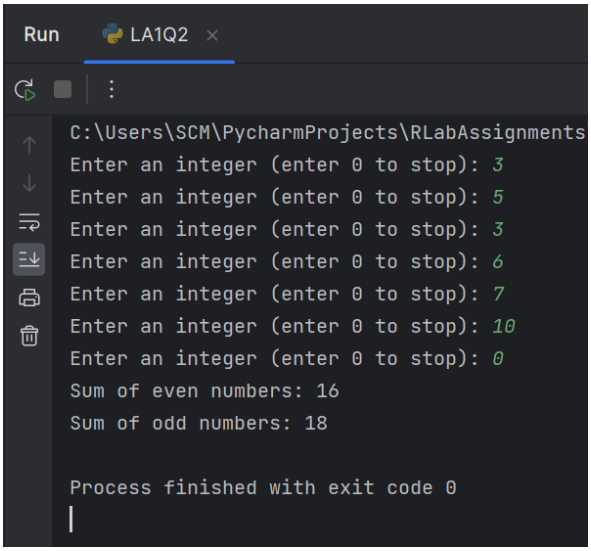
Answer:

Code:

```
sum_even = 0
sum_odd = 0

while True:
    user_input = input("Enter an integer (enter 0 to stop): ")

    if user_input.isdigit():
        number = int(user_input)

        if number == 0:
            break  # Exit the loop if 0 is entered

        if number % 2 == 0:
            sum_even += number
        else:
            sum_odd += number
```

```
    else:
        print("Please enter a valid integer.")

print("Sum of even numbers:", sum_even)
print("Sum of odd numbers:", sum_odd)
```

Output:



Question No. 3:

Fibonacci series is that when you add the previous two numbers the next number is formed. You
have to start from 0 and 1.
E.g. 0+1=1 → 1+1=2 → 1+2=3 → 2+3=5 → 3+5=8 → 5+8=13
So the series becomes
0 1 1 2 3 5 8 13 21 34 55 ……
Steps: You have to take an input number that shows how many terms to be displayed. Then use loops for displaying the Fibonacci series up to that term e.g. input no is =6 the output should be
0 1 1 2 3 5

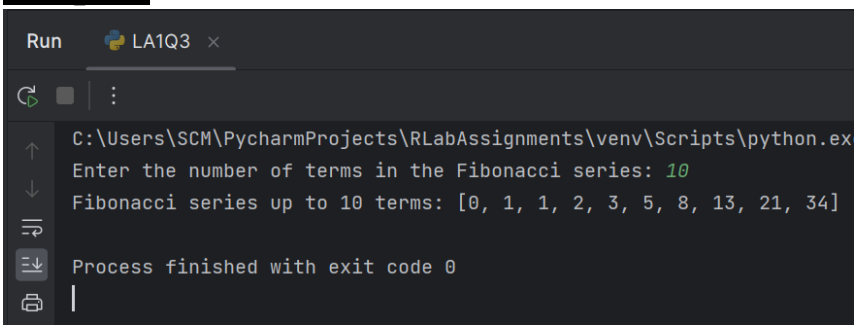
Answer:

Code:

```
def generate_fibonacci(n):
    fibonacci_series = [0, 1]

    for _ in range(2, n):
        next_term = fibonacci_series[-1] + fibonacci_series[-2]
        fibonacci_series.append(next_term)

    return fibonacci_series

num_terms = int(input("Enter the number of terms in the Fibonacci series: "))

if num_terms <= 0:
    print("Please enter a positive integer.")
else:
```

```
    fibonacci_result = generate_fibonacci(num_terms)
    print("Fibonacci series up to", num_terms, "terms:", fibonacci_result)
```

Output:



Question No. 4:

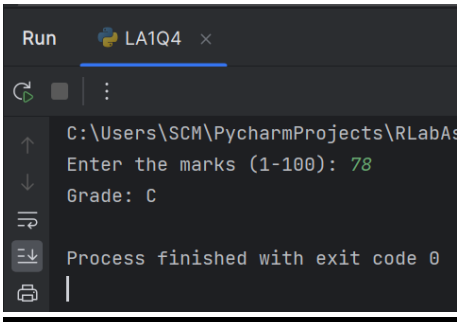
Write a Python code to accept marks of a student from 1-100 and display the grade according to the following formula.
Grade F if marks are less than 50
Grade E if marks are between 50 to 60
Grade D if marks are between 61 to 70
Grade C if marks are between 71 to 80
Grade B if marks are between 81 to 90
Grade A if marks are between 91 to 100

Answer:

Code:

```
marks = float(input("Enter the marks (1-100): "))

if 0 <= marks <= 100:
    if marks < 50:
        grade = "F"
    elif 50 <= marks <= 60:
        grade = "E"
    elif 61 <= marks <= 70:
        grade = "D"
    elif 71 <= marks <= 80:
        grade = "C"
    elif 81 <= marks <= 90:
        grade = "B"
    else:
        grade = "A"

    print(f"Grade: {grade}")
else:
    print("Please enter a valid marks value between 0 and 100.")
```

Output:



Question No. 5:

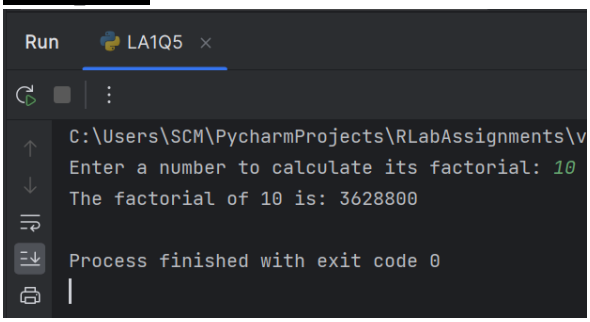
Write a program that takes a number from user and calculate the factorial of that number.

Answer:

Code:

```
def calculate_factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * calculate_factorial(n - 1)

number = int(input("Enter a number to calculate its factorial: "))

if number < 0:
    print("Please enter a non-negative integer.")
else:
    factorial_result = calculate_factorial(number)
    print("The factorial of", number, "is:", factorial_result)
```

Output:



Question No. 6:

Write a program (function) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates.

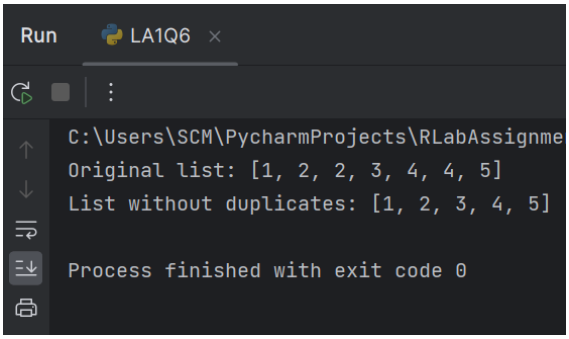
Answer:

Code:

```
def remove_duplicates(input_list):
    unique_list = []
```

```
    for item in input_list:
        if item not in unique_list:
            unique_list.append(item)

    return unique_list


input_list = [1, 2, 2, 3, 4, 4, 5]
result_list = remove_duplicates(input_list)

print("Original list:", input_list)
print("List without duplicates:", result_list)
```

Output:



Question No. 7:

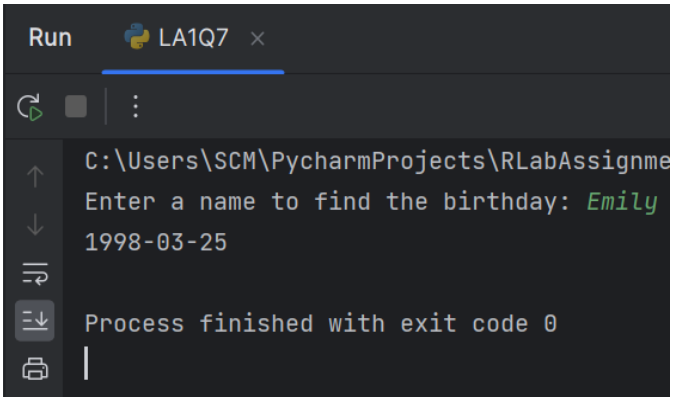
For this exercise, you will keep track of when our friend's birthdays are, and be able to find that information based on their name. Create a dictionary (in your file) of names and birthdays. When you run your program it should ask the user to enter a name, and return the birthday of that person back to them.

Answer:

Code:

```
birthdays_dict = {
    "John": "1990-05-15",
    "Alice": "1985-12-20",
    "Bob": "1993-08-10",
    "Emily": "1998-03-25",
    "Charlie": "1980-11-05"
}

def find_birthday(name):
    return birthdays_dict.get(name, "Birthday not found")

user_input = input("Enter a name to find the birthday: ")

result = find_birthday(user_input)
print(result)
```

Output:



Question No. 8:

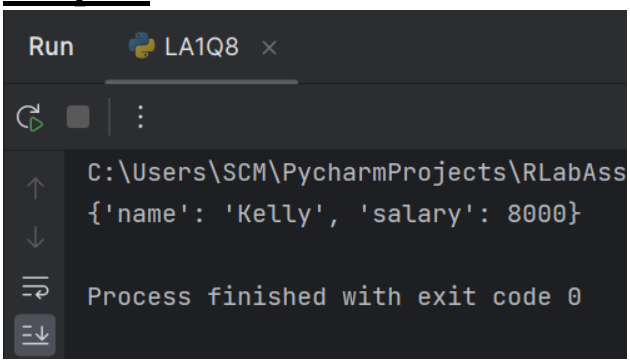
Create a dictionary by extracting the keys from a given dictionary
Write a Python program to create a new dictionary by extracting the mentioned keys from the below dictionary.
Given dictionary:
sample_dict = { "name": "Kelly", "age": 25, "salary": 8000, "city": "New york"}
 # Keys to extract
keys = ["name", "salary"]
Expected output:
{'name': 'Kelly', 'salary': 8000}

Answer:

Code:

```
sample_dict = {"name": "Kelly", "age": 25, "salary": 8000, "city": "New York"}

keys_to_extract = ["name", "salary"]

extracted_dict = {key: sample_dict[key] for key in keys_to_extract if key in
sample_dict}

print(extracted_dict)
```

Output:



Question No. 9:

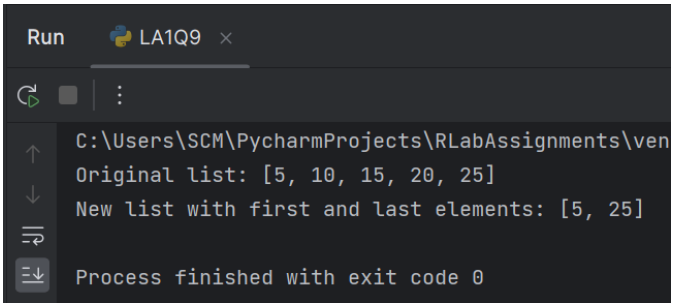
Write a program that takes a list of numbers (for example, a = [5, 10, 15, 20, 25]) and makes a new list of only the first and last elements of the given list. For practice, write this code inside a function.

Answer:

Code:

```
def first_and_last_elements(input_list):
    if len(input_list) >= 2:
        return [input_list[0], input_list[-1]]
    else:
        return "List should have at least two elements"

a = [5, 10, 15, 20, 25]
result_list = first_and_last_elements(a)

print("Original list:", a)
print("New list with first and last elements:", result_list)
```

Output:



Question No. 10:

Write a combined program showing all the collections in a specific scenario.

Answer:

Code:

```
def main():
    students = {
        "John": {"age": 20, "courses": ["Math", "Physics"]},
        "Alice": {"age": 22, "courses": ["Chemistry", "Biology"]},
        "Bob": {"age": 21, "courses": ["History", "Literature"]}
    }

    print("Initial Student Database:")
    display_student_database(students)

    add_student(students, "Emily", 19, ["Computer Science"])

    update_student_courses(students, "John", ["Math", "Physics", "Computer
Science"])

    print("\nUpdated Student Database:")
```

```
    display_student_database(students)

def display_student_database(database):
    for name, info in database.items():
        print("Name: {}, Age: {}, Courses: {}".format(name, info['age'],
info['courses']))

def add_student(database, name, age, courses):
    database[name] = {"age": age, "courses": courses}
    print("\n{} has been added to the database.".format(name))

def update_student_courses(database, name, new_courses):
    if name in database:
        database[name]["courses"] = new_courses
        print("\n{}'s courses have been updated.".format(name))
    else:
        print("\n{} not found in the database. Unable to update
courses.".format(name))

if __name__ == "__main__":
    main()
```

Output: